Project Milestone 2 (Common Sense Validation)

Aarshee Mishra aarsheemishr@umass
Anupam Yadav anupamyadav@umass
Sangeetha Balasubramanian sangeethabal@umass
Sumeha Kashyap sumehakashya@umass

1 Introduction

Natural Language sentences can often be syntactically and grammatically correct but may not make any sense. Common sense reasoning is one of the main bottle necks in machine intelligence. Research on common sense understanding systems has received increased attention. In our work we discuss a baseline approach and experiment with three new approaches to differentiate natural language statements that make sense from those that do not make sense.

2 Your dataset

For our task, given a pair of natural language statements we identify which of them makes more sense. Both sentences have similar syntactic structures differing only by a few words. Some examples of sentences that don't make sense are: *'He poured orange juice on his cereal', 'He drinks apple', 'Jeff ran 100,000 miles today'*. While corresponding sentences that make sense are: *'He poured milk on his cereal', 'He drinks milk', 'Jeff ran a mile today'*.

In our dataset, we have a total of 12021 pairs of sentences. The entire data has been annotated and is available online[1]. The dataset(Wang et al., 2019) has been released as part of the SemEval 2020 contest. We refer to the first sentence as *sent0* and second one as *sent1*.

The total number of words in our vocabulary is 9018. The average length of the sentences is 39.3 ± 13.6. The distribution of sentence lengths of the two classes is shown in figure 1. The average sentence length difference between the two classes is 2.8 ± 3.1. Figure 2 shows the absolute difference in word count between two classes in decreasing order. In our test set, we have 2021 sentence pairs. In 1352 pairs both the sentences have the same length and differ by just one word, occurring in the same place in both sentences.

[1] https://github.com/wangcunxiang/Sen-Making-and-Explanation

Model	Accuracy (%)
Naive Bayes (Baseline)	56.9
BERT (Sentence Pair)	88.71
BERT (Single Sentence)	71.25
BERT (Wang et al., 2019)	70.1

Table 1: Shows the results for the different models used for the classification task.

3 Baselines

We have one baseline model for our dataset which is a naive-bayes classifier. We pre-process the input text using white space tokenization to create a bag-of-unigrams representation. The dataset is split into train and test sets in the ratio of 83:17. This results in 10000 training examples and 2021 test examples. The training set is split further to include a validation set of 2000 examples (20% of the training data). Our evaluation metric is accuracy which is determined by the ratio of 'number of sentence pairs in which both the sentences are correctly classified' to 'the total number of sentence pairs'. The hyperparameter for our baseline model is the smoothing parameter α. Tuning the parameter α, we observe that the accuracy for the validation set increases sharply for small values of alpha and then dips gradually as alpha increases. For $\alpha = 2$ we observe the test accuracy to be 56.9%

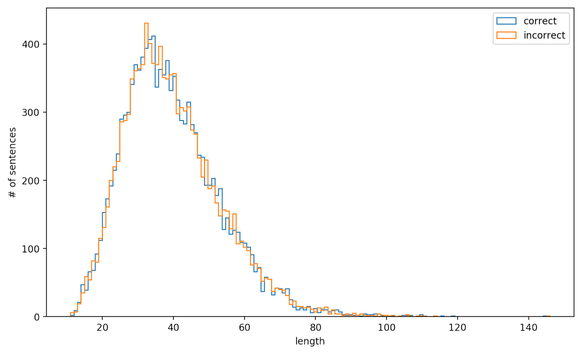

Figure 1: Distribution of sentence lengths.

4 Your approach

We have implemented and evaluated 3 approaches for our task. The best approach gives an accuracy of 88.80%.

Our first approach is inspired by the fact that almost fifty percent of the examples in the dataset have the same length for *sent0* and *sent1* and differ by only one token. This approach is implemented only on the sentences which follow the above rule. For every sentence pair, we mask out the word that is different and predict the probability distribution of this MASK using a pre-trained BERT(Devlin et al., 2018) language model. Using this distribution, we calculate the probability of the actual word. The sentence which has a higher probability for the actual word is classified as 'makes sense' while the other as 'does not make sense'.

In the second approach, we fine-tune BERT for Sequence Classification. For every training example, we use *sent0* as the first sentence input and *sent1* as the second sentence input. The targets for the classification task are binary labels. The label is '1' when the *sent0* is the sentence that 'makes sense' and '0' otherwise. We tuned the initial learning rate and the total number of epochs to make the model converge. On using the default BERT parameters ('weight-decay': 0, 'learning-rate': 4e-5,'adam-epsilon': 1e-8,'train-batch-size': 8, warm-up steps :0) and setting total epochs to 1, causes the loss to hover around the initial value after the epoch is over. However changing the total epochs to 10, made the loss fall after 1 epoch. We realized that this happened because, traditionally BERT uses a Linear Scheduler, and the rate of fall of the learning rate depends on the total number of epochs. Thus setting the total epochs to a larger number, makes the fall of learning rate less extreme and helps in model convergence. We assume that the model converges when the training loss stops changing. This model gives an accuracy of 88.71%.

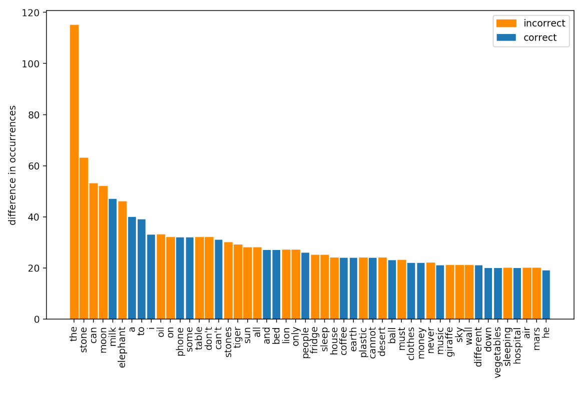

Figure 2: Absolute difference in word count.

For our third approach, we again fine-tune BERT for Sequence Classification but change the input format of the model. We do not use the second input to BERT. For every example in the dataset, we used both *sent0* and *sent1* as the first input to BERT in an iterative fashion. The targets for the classification task are binary labels. The label is '1' when the input 'makes sense' and '0' when the input 'does not make sense'. We shuffle the dataset so that sentences that belong to the same dataset example do not always occur one after the other. As in the previous approach, we tuned the initial learning rate and the total epochs to achieve convergence. This model gives an accuracy of 71.25%.

5 Error analysis

Table1 and Table 2 show the results of our approaches.

1. Bert Language Model -
   Examples of incorrectly classified sentence pairs of the same length that differ by one word-
   (a) She drove her children to the moon, She drove her children to the playground
   (b) I was playing basketball with a ball, I was playing basketball with a rock

   Both the sentences here are pretty obvious and a person would have no difficulty in detecting the more logical statement.

2. Bert for sentence pair classification -
   Examples of incorrectly classified sentence pairs of the same length that differ by one word-
   (a) The car is driving slowly on the highway , The car is driving fast on the highway
   (b) A boy is a male , A girl is a male

   Examples of incorrectly classified sentence pairs that differ in more than one word -
   (a) Animals are pets, Pets are animals
   (b) Vegetarian usually like eating meat, Vegetarian usually don't eat meat

Model	Accuracy (%)
BERT (Sentence Pairs)(a)	88.53
BERT (Sentence Pairs)(b)	94.768
BERT (Single Sentence)(a)	71.67
BERT (Single Sentence)(b)	70.403
BERT (Masked LM)(a)	77.27

Table 2: Shows accuracy of models on sentence pairs of the same length differing by one word and sentence pairs that differ by more than a word. a) Sentence pairs that have the same length and differ by one word at a particular position in the sentence. b) Sentence pairs that differ by more than a word.

While some of the misclassified sentence pairs are actually ambiguous and require more context (The car is driving slowly on the highway, The car is driving fast on the highway), there are some trivial sentence pairs (A boy is a male, A girl is a male) as well that are getting misclassified.

3. BERT for Single Sentence Classification - Examples of incorrectly classified sentence pairs that differ by only one word:

   (a) He walks in from the wall, He walks in from the door

   (b) People usually work on Sundays, People usually rest on Sundays

   Examples of incorrectly classified sentence pairs with different lengths:

   (a) I cooked my meal at the restaurant, I paid for my meal at the restaurant

   (b) A lemon tastes sour when it goes bad, Milk tastes sour when it goes bad

6 Timeline for the rest of the project

1. Incorporate external knowledge from various knowledge bases like ConceptNet to word embeddings. (Nov 22 - Nov 28)

2. Analyze performance of our existing models with ConceptNet+BERT (Nov 28 - Dec 01)

3. Final report and presentation (Dec 02 - Dec 10)

References

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Wang, C., Liang, S., Zhang, Y., Li, X., and Gao, T. (2019). Does it make sense? and why? a pilot study for sense making and explanation. *arXiv preprint arXiv:1906.00363*.